

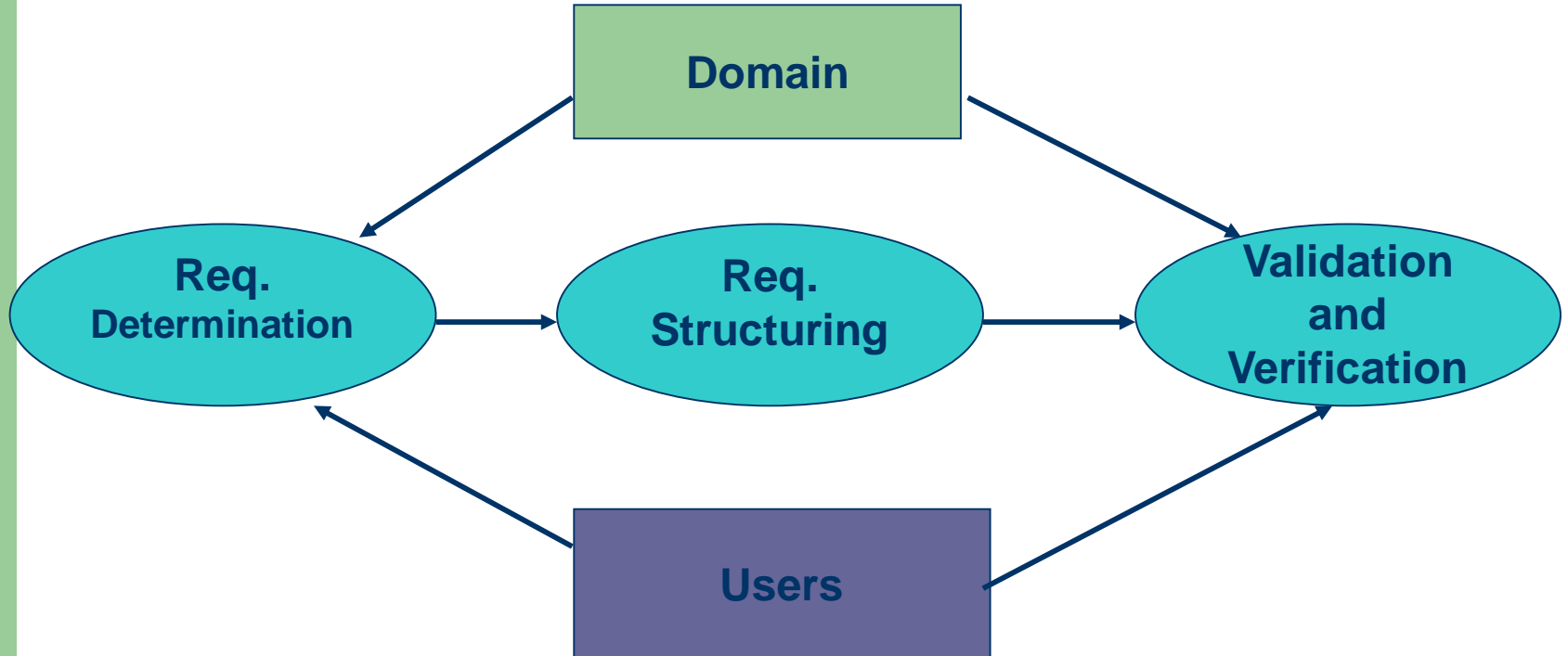
Chapter 4

Determining System Requirements



Requirement Engineering Process

- The process of determining the services that the customer requires from a system and the constraints under which it operates and is developed
- **The requirements** are the descriptions of the system services and constraints that are generated during the requirements engineering process
- It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification



1- Performing Requirements Determination (Elicitation)

- **Gather information on what system should do from many sources**
 - **Users: Interviews, questionnaire, forms,**
 - **Documents (Domain): Reports, Procedures, previous systems, domain**
 - **Brain storming sessions**
- **Note that making SW for specific users, elicitation is different from making generic SW. ???**

Deliverables and Outcomes

- Types of deliverables:
 - Information collected from users
 - Existing documents and files
 - Computer-based information
 - Understanding of organizational components
 - Business objective
 - Information needs
 - Rules of data processing

Performing Requirements Determination

- **Characteristics for gathering requirements**

- Question everything (Ex. Are all the transactions proceeded in the same way)
- Impartiality
 - Find the best organizational solution. Consider all ideas and try to find the best.
- Relaxation of constraints Assume everything is possible and eliminate the infeasible.
- Attention to details
- Reframing
 - View the organization in new ways according to the users requirements

Traditional Methods for Determining Requirements

1- Interviewing and Listening

- Gather facts, opinions,...
- Observe body language and emotions (feeling)
- **Guidelines**
 - **Make a Plan**
 - **Appointment (convenient for the interviewee)**
 - Give him an idea of the subject of interview (he may need to prepare things before)
 - Checklist of the interview
 - **Be neutral**
 - **Listen carefully**

Traditional Methods for Determining Requirements

- Type up notes within 48 hours
- Do not set expectations about the new system

- **Interview Questions**

- Open-Ended
 - No pre-specified answers (gives the interviewee freedom to speak)
- Close-Ended
 - Interviewee is asked to choose from a set of specified responses (T/F or MCQ or ranking)
- Do not phrase questions in ways that imply a wrong or right answer

Traditional Methods for Determining Requirements

2- Questionnaires

- More cost-effective than interviews
- Be careful when Choosing respondents
 - Should be representative of all users
- Design
 - Mostly closed-ended questions
 - Can be administered over the phone or in person

Traditional Methods for Determining Requirements

3- Interviewing Groups

- Make an interview of a group instead of individually
- Advantages
 - More effective use of time
 - Enables people to hear opinions of others and to agree or disagree
- Disadvantages
 - Difficulty in scheduling a specific time

Traditional Methods for Determining Requirements

4- Directly Observing Users

- Observe the users while they are working to understand how do they perform the work
- Serves as a good method to supplement interviews

- **Disadvantage:**

- Often difficult to obtain unbiased data as people often work differently when being observed (better or worse)

Analyzing Procedures and Other Documents

- Four types of useful documents
 - Written work procedures
 - Describes how a job is performed
 - Includes data and information used and created in the process of performing the job or task
 - Business form
 - Explicitly indicate data flow in or out of a system
 - Report
 - Enables the analyst to work backwards from the report to the data that generated it
 - Description of current information system



**SOFTWARE
SELECTIONS, INC.**
1234 COMPUTER CT.
ANY TOWN, ANY STATE 54321
(123) 456-7890

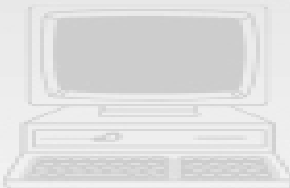
INVOICE

INVOICE NO.

SOLD
TO

SHIP
TO

| ACCOUNT NO. | ORDER NO. | PURCHASE ORDER NO. | SHIP VIA | COLL. PYD. | DATE SHIPPED | TERMS | INVOICE DATE |
|-------------|--------------|--------------------|----------|---------------|--------------|-------|--------------|
|-------------|--------------|--------------------|----------|---------------|--------------|-------|--------------|

| QTY. ORDERED | QTY. SHIPPED | QTY. B.C. | ITEM NO. | DESCRIPTION | UNIT PRICE | MISC. CHG. | EXTENDED PRICE |
|-----------------|-----------------|--------------|----------|--|------------|---------------|----------------|
| | | | |  | | | |

| | | |
|--|---------------------------------------|--|
| | SALE AMOUNT | |
| | MISC. CHARGES SALES TAX FREIGHT | |
| | TOTAL | |

FORM ST2002

Financial Highlights

| | 1999 | 1998 | 1997 | 1996 | 1995 |
|--|-----------|-----------|-----------|-----------|------------|
| Net Sales | \$4,741.1 | \$3,831.9 | \$3,831.2 | \$3,327.5 | \$2,900.1 |
| Income from continuing operations | \$ 511.0 | \$ 354.8 | \$ 430.8 | \$ 355.2 | \$ (130.3) |
| Income from discontinued operations, net of tax | 4.8 | 88.5 | 30.0 | (138.0) | (20.0) |
| Extraordinary charge, net of tax and minority interest | | | | (0.9) | (1.8) |
| Net Income | \$ 515.8 | \$ 421.3 | \$ 481.5 | \$ 217.4 | \$ (102.9) |

Diluted Earnings Per Share

| | | | | | |
|-------------------------|---------|---------|---------|---------|-----------|
| Continuing operations | \$ 1.95 | \$ 1.42 | \$ 1.71 | \$ 1.44 | \$ (0.49) |
| Discontinued operations | 0.02 | 0.25 | 0.11 | (0.54) | 0.11 |
| Net Income | \$ 1.97 | \$ 1.67 | \$ 1.82 | \$ 0.90 | \$ (0.38) |

Net Sales
(in millions)

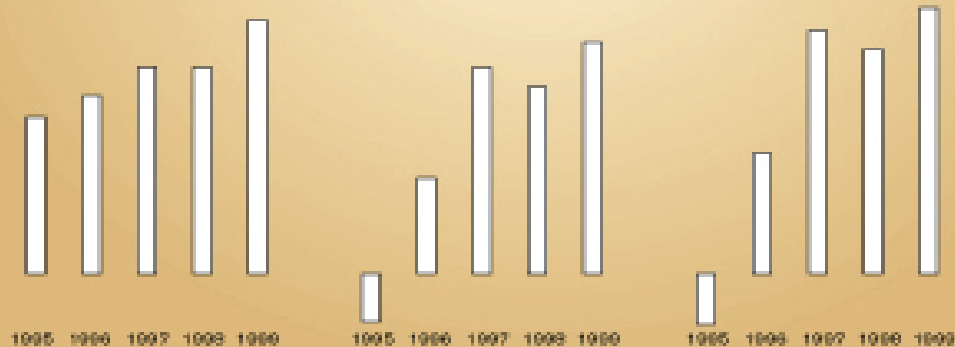
\$2,900 \$3,328 \$3,831 \$3,832 \$4,741

Net Income
(in millions)

\$ (102) \$217 \$482 \$421 \$516

Diluted Earnings Per Share

\$ (0.38) \$0.90 \$1.82 \$1.67 \$1.97



Example of a report

Analyzing Procedures and Other Documents

- Types of information to be discovered:
 - Problems with the existing system
 - Opportunity to meet new need
 - Organizational direction
 - Names of key individuals
 - Values of organization
 - Special information processing circumstances
 - Reasons for current system design
 - Rules for processing data

Modern Methods for Determining Requirements

1) Joint Application Design (JAD)

- Brings together key users, managers and systems analysts
- Purpose: collect system requirements simultaneously from key people
- **End Result of JAD**
 - Documentation detailing existing system
 - Features of proposed system

2) Prototyping

- Repetitive process
- Rudimentary version of system is built
- Replaces or augments SDLC
- Goal: to develop concrete specifications for ultimate system

Prototyping

- Quickly converts requirements to working version of system
- Once the user sees requirements converted to system, will ask for modifications or will generate additional requests
- Most useful when:
 - User requests are not clear
 - Few users are involved in the system
 - Designs are complex and require concrete form
 - History of communication problems between analysts and users
 - Tools are readily available to build prototype

Prototyping

- **Drawbacks**

- Tendency to avoid formal documentation
- Difficult to adapt to more general user audience
- Sharing data with other systems is often not considered

(V&V)

Verification:

- are we building *the thing right?* (*the way of building itself*)
- we wrote every details, no errors in spelling, or something missing.

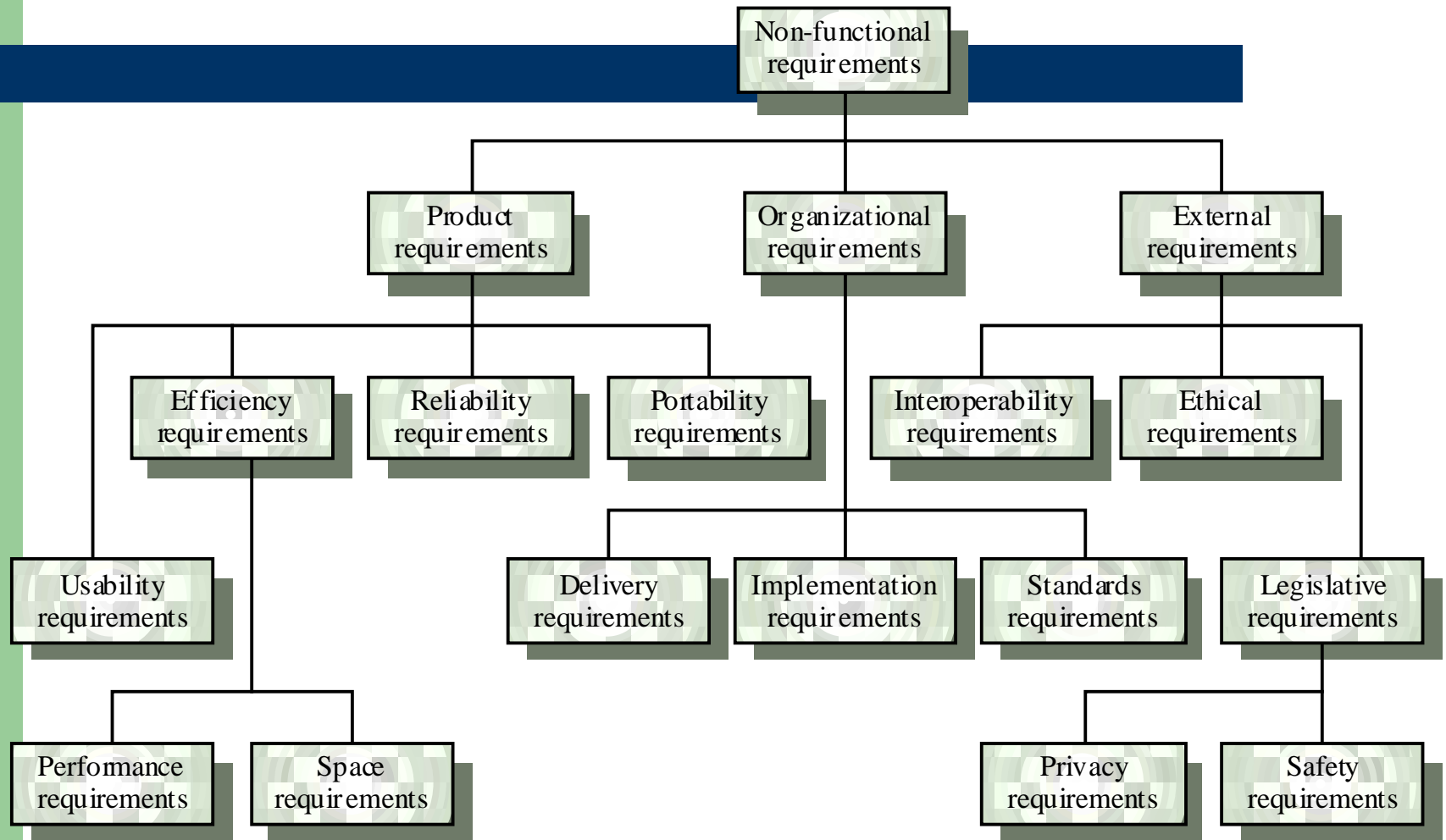
Validation:

- are we building *the right thing?* (*the thing itself*)
- is each requirement written as it should be programmed or is it clear (not understandable).

Functional and non-functional Requirements

- **Functional requirements**
It describes the functions (services) required, how the system should react to particular inputs and how the system should behave in particular situations.
- **Non-Functional requirements**
- Define system properties (e.g. reliability, response time and storage requirements) and constraints(e.g. I/O device capability, system representations, etc.). It may also specify a particular programming language or development method.
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless

Non-functional requirements Classification



Examples of Non-functional requirements

| Property | Measure |
|-------------|--|
| Speed | Processed transactions/second User/Event response time Screen refresh time |
| Size | K Bytes Number of RAM chips |
| Ease of use | Training time Number of help frames |
| Reliability | Mean time to failure Probability of unavailability Rate of failure occurrence Availability |
| Robustness | Time to restart after failure Percentage of events causing failure Probability of data corruption on failure |
| Portability | Percentage of target dependent statements Number of target systems |

Non-functional requirements

1- Product requirements

- Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, portability, etc.

2- Organisational requirements

- Requirements which are driven from organisational policies and procedures e.g. process standards used, implementation requirements such as colours, fonts used, number of users, security measures, delivery date, etc.

3- External requirements

- Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements with other programs, legislative requirements, etc.